

## LOCATION BASED ROUTING FOR MOBILE AD-HOC NETWORKS

### CROSS-REFERENCE TO RELATED APPLICATIONS

sub 5  
a1  
Related subject matter is disclosed in the co-pending, commonly assigned, U.S. Patent application of Ahmed et al., entitled "A Topology Sensing Scheme for Networks with Mobile Nodes," Application No. XXX, filed on XXX.

This invention was made with Government support under contract DAAB07-98-C-D009. The Government has certain rights in this invention.

### FIELD OF THE INVENTION

This invention relates generally to communications and, more particularly, to wireless systems.

### BACKGROUND OF THE INVENTION

An "ad-hoc" mobile network (ad-hoc network) is a wireless network that comprises a collection of nodes whose positions are continually changing. Unlike a regular wireless network, one can view an ad-hoc network as a network with no fixed infrastructure. For example, all the nodes function as routers and perhaps as base stations; and the mobility of the nodes causes frequent changes in network topology.

It is the varying network topology of an ad-hoc network that causes difficulty in applying routing techniques used in a conventional wireless network. In the latter, the nodes in the network are stationary and the links connecting the nodes go down infrequently. As such, it is possible to maintain the whole network topology at each node by sending topology-related information to all the nodes in the network via, what is known in the art as, "link-state," updates. Since nodes go down infrequently - link-state updates are infrequent - and this approach works quite well in a conventional wireless network. However, in an ad-hoc network link-state changes are more frequent because of the shifting topology, thus generating many more link-state update messages throughout the ad-hoc network - and consuming valuable bandwidth in the process. Also, construction of consistent routing tables is difficult because of the delay involved in propagating link-

state information.

Considering these factors, routing protocols for ad-hoc networks can be classified broadly into two categories: “table-driven” and “source initiated on-demand.” Table-driven routing protocols are similar to the above-mentioned conventional wireless routing approach, i.e., each node attempts to maintain consistent, up-to-date, routing information for all other nodes in the network. Examples of table driven routing protocols are “Destination-Sequenced-Distance-Vector” (DSDV), “Clusterhead Gateway Switch Routing” (CGSR), and the “Wireless Routing Protocol” (WRP) protocols. In contrast, source initiated on-demand routing protocols create routing information only when a source node needs a route to a given destination. Examples of source initiated on-demand routing protocols include “Ad-Hoc On-Demand Distance Vector” (AODV), “Dynamic Source Routing” (DSR), “Temporally Ordered Routing Algorithm” (TORA), and the “Zone Routing Protocol” (ZRP) protocol.

As an illustration of a source initiated on-demand protocol consider ZRP. In ZRP, each node maintains the whole network topology for a local area, or zone, around it. As such, if the node (i.e., the source node) has to send a packet to a destination address in the zone, that routing information is already available. However, if the source node has to send a packet to a destination address outside their zone, then the node initiates a query to all the nodes in the edge of its zone (i.e., edge nodes). If one of these edge nodes has the routing information for the destination address, then that routing information is passed on back to the source node.

## SUMMARY OF THE INVENTION

We have observed that the above-mentioned forms of ad-hoc network routing protocols generally require a node maintaining accurate information, in one form or another, about how to route to a node in regions that are far away from it. As such, if the number of nodes is large and if there is reasonable mobility of the nodes, getting this information becomes difficult — if not impractical. Therefore, we have devised a geometry-based routing protocol (GRP) in which a source nodes routes a packet to a destination node outside of its local node topology (referred to herein as the local topology) as a function of the distance to the location of a destination node. In

accordance with the invention, each node stores location information for nodes of the ad-hoc network and exchanges this location information with adjacent nodes. Thus, location information is gradually propagated throughout the ad-hoc network.

In an embodiment of the invention, each node of an ad-hoc network maintains a location list, which comprises location information for a number of nodes of the ad-hoc network. Periodically, each node transmits to its direct neighbors (i.e., those nodes with which it has a point-to-point link) (a) its location, and (b) its location list. Alternatively, each node periodically transmits its location to all nodes in its local topology and its location list to all of its direct neighbors. Each node that receives a location list from an adjacent node merges the received location list into its own location list such that location information for existing nodes, and/or newly identified nodes, is current.

## BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 shows a portion of an ad-hoc network embodying the principles of the invention;

FIG. 2 shows an illustrative local topology table;

FIG. 3 shows an illustrative location table;

FIG. 4 shows an illustrative flow chart for use in routing a packet in an ad-hoc network;

FIG. 5 shows an illustrative routing table;

FIG. 6 shows an illustrative 2-region for node 105 of FIG. 1;

FIG. 7 shows an illustrative flow chart for use in constructing a local topology;

FIG. 8 shows an illustrative direct neighbor table;

FIG. 9 shows another illustrative location table;

FIG. 10 shows an illustrative flow chart for use in a lazy update procedure; and

FIG. 11 shows an illustrative high-level block diagram of a node for use in the ad-hoc network of FIG. 1.

## DETAILED DESCRIPTION

A portion of an illustrative ad-hoc network embodying the principles of the invention is shown in FIG. 1. Other than the inventive concept, the elements shown in

FIG. 1 are well-known and will not be described in detail. For example, node 105 includes stored-program-control processors, memory, and appropriate interface cards for wireless communications. (The exact form of wireless transmission used, e.g., the use of carrier-division multiple access (CDMA), is not relevant to the inventive concept and, as such, is not described herein.) For the purposes of this example, it is assumed that each node of the ad-hoc network refers to a mobile device that allows users (mobile user stations, terminals, etc. (not shown)) to access the ad-hoc network and also provides routing functions for packets/data traversing the network. Each node transmits an omnidirectional pilot signal and is capable of communicating with other nodes using a signaling protocol to transfer information, such as the earlier-mentioned link-state information, between nodes. (Pilots and signaling protocols are known in the art and, as such, are not described herein.) The omnidirectional antenna and pilot signal are part of a topology sensing scheme (referred to further below) which enables nodes to sense the presence of one another and also to exchange some information useful for making link setup decisions. In general, and other than the below-described inventive concept, the nodes use this information to decide which of their neighboring nodes they should have direct (point-to-point) links with and then proceed to establish these links. The point-to-point links are preferably supported by directional antennas.

For the sake of simplicity it is assumed that all nodes with a transmission radius,  $r$ , of node 105 are capable of communicating with node 105.

At this point, the following definitions are made:

$V$  – represents the set of all nodes in the ad-hoc network;

$v, w, u, i, j$  – represent various nodes of the ad-hoc network;

$r$  – transmission radius for a node, i.e., all nodes within the transmission radius are capable of communicating with that node;

$N(v)$  – represents the local topology of a node,  $v$ ;

$S^k(v)$  – the  $k$ -neighborhood of a node  $v$ ; i.e., a local topology of node  $v$  where all nodes are within  $k$  hops of node  $v$ ;

$H_{vw}$  – the minimum number of hops between a node  $v$  and a node  $w$ , where  $w \in N(v)$ ;

$N_{vw}$  – the next hop node from a node  $v$  to a node  $w$ , where  $w \in N(v)$ ;

$l(v)$  – represents the location of a node  $v$ ; and

$D_{vw}$  – represents the distance between two nodes,  $v$  and  $w$ ; where

$$D_{vw} = \|l(v) - l(w)\|. \quad (1)$$

It is assumed that each node further comprises global positioning system (GPS) equipment (not shown in FIG. 1), as known in the art, for determining its own location (in two dimensions) on the globe. In accordance with the invention, each node of the ad-hoc network implements a geometry-based routing protocol (GRP) (also referred to as a geometry-based routing algorithm (GRA) or position-based routing) such that:

(a) each node has its own defined local topology, (also referred to as a local network or a local neighborhood) (described further below) which may, or may not, be different than the local topologies of other nodes; and

(b) each node stores location information (approximate or exact) of the nodes of the ad-hoc network (those nodes in the local topology and those nodes outside of, or distant from, the local topology).

In other words, in the GRP each node knows its local topology for a subset of nodes of the ad-hoc network (connectivity and location) and only location information for other, or distant, nodes of the ad-hoc network (i.e., connectivity is not known for these distant nodes). As will become apparent from the description below, the GRP is capable of implementation using conventional programming techniques, which, as such, will not be described herein.

Illustratively, FIG. 1 shows a local topology 100 for node 105. As can be observed from FIG. 1, local topology 100 not only defines the nodes that are a part of local topology 100 but also how node 105 is connected to these nodes (i.e., a “network graph,” or simply “graph”). It is assumed that all communications are bi-directional and hence the graph is undirected; and that local topology 100 is non-hierarchical. Illustratively, node 105 stores in memory (not shown) a local topology table (as illustrated in FIG. 2), which corresponds to local topology 100 and a location table (as illustrated in FIG. 3), which stores location information for nodes (including nodes outside the local

topology). As defined above, local topology 100 is representative of a 2-neighborhood for node 105, i.e.,  $S^2(105)$ , since all nodes of local topology 100 can be reached from node 105 in 2, or fewer, hops. As used herein, node 105 is the reference node for local topology 100.

5 As noted, the local topology table of FIG. 2 lists all the nodes currently in the local topology for node 105 and the connection between nodes. For example, if node 105 has a packet to transmit to node 115, node 105 transmits the packet to the next hop node, which is identified as node 110 from the local topology table. From this table, the total number of hops to get to node 115 is  $k = 2$ . This is also illustrated in FIG. 1 by arrow 101. Creation of the local topology table is described further below.

Although node 105 is capable of communicating with all nodes within the transmission radius  $r$ , node 105 only communicates with nodes with which it has established point-to-point links (i.e., its direct neighbors). Similarly, other nodes only communicate with node 105 if node 105 is their direct neighbor. In other words, nodes 15 are preferably connected as point-to-point wireless links that gives rise to a  $k$ -neighborhood for a node, which is referred to herein as the local topology for that node. (Also, as noted above, it is possible to use directional antennae and focused beams to communicate between the neighbors in the graph — thereby increasing the capacity of the system.)

20 Since each node has its own local topology and location information for nodes (including those outside its local topology), the GRA is defined as follows. Let  $t$  be the destination address (of a destination, or end, node) of a packet that arrives at a node  $v$ , which has a local topology,  $N(v)$ . In accordance with the GRA, if  $t \neq v$ , node  $v$  determines:

25 
$$w = \arg \min_{u \in S^1(v)} D_{ut} ; \quad (2)$$

where node  $v$  forwards the packet to node  $N_{vw}$  unless  $w = v$  (i.e., the reference node itself is the closest node) in which case the packet is dropped.

Using the GRA, the packet, in effect, migrates from local topology to local 30 topology until reaching that local topology within which the end node resides.

In the context of FIG. 1, the GRA is illustrated as follows, and as is shown in the flow chart of FIG. 4. Assume that node 105 (the source node,  $v$ , of equation (2)) receives a packet (not shown) for transmission to node 205 (the end node,  $t$ , of equation (2)) in step 405 of FIG. 4. Node 105 searches its local topology table to see if node 205 is a part of its local topology in step 410. If it is, node 105 simply sends the packet to the next hop node identified in its local topology table in step 415. On the other hand if node 205 is not a part of the local topology for node 105, node 105 performs the geometry-based routing protocol in step 420 to identify the closest node, in its local topology, to node 205. In particular, node 105 performs equation (2) for all nodes that are a part of its local topology 100. Node 105 evaluates the distance from node 205 to each node in its local topology 100 (using equation (1) and the location information from the location table shown in FIG. 3). This is illustrated in FIG. 1 by three dotted line arrows  $D_{140, 205}$ ,  $D_{105, 205}$ , and  $D_{150, 205}$ , which correspond to the distance calculations of equation (1) between nodes 140 and 205, 105 and 205, and 150 and 205 (the other distance calculations for the remaining nodes of local topology 100 are not shown). Once the closest node is identified, node 105 sends the packet to that node of local topology 100 that has the minimum distance to node 205, e.g., here assumed to be node 140. Node 105 routes the packet to node 140 via the local topology table, in step 415 of FIG. 4 (i.e., the packet is sent to the next hop node 130 as indicated in the local topology table of FIG. 2). As should be readily apparent, the next hop node then performs the GRA using its local topology table. (Although not shown in the flow chart of FIG. 4, suitable error conditions can also be added to process the packet in certain situations. For example, if there is no location information for node 205 in the location table, the packet is dropped.)

In the application of the GRA within a local topology it is important to ensure that there are no “loops” in the routing. One possible cause of a loop in the GRA routing is the situation where two nodes are the same distance from the destination node. As such, an alternative to equation (2) is equation (3), below:

$$w = \arg \min_{u \in S^k(v)} D_{ut} + \epsilon H_{vu}; \quad (3)$$

where it is assumed that  $\epsilon$  is a very small number. The implication of  $\epsilon$  is that if there are two nodes whose distance to the end node is the same, then the tie is broken in

favor of the node that is closer to  $u$  in terms of hop count.

Also, rather than making routing calculations on-the-fly as packets arrive at a node as illustrated in FIG. 4, a routing table can be constructed *a priori* using the calculations described earlier and packet routing decisions can be made on the basis of the entries in the routing table. Such an illustrative routing table is shown in FIG. 5. This routing table uses the information from both the local topology table illustrated in FIG. 2 and the location table illustrated in FIG. 3, along with the above-described routing calculations (e.g., equation (3)). Using the same example above, and as illustrated in FIG. 5, a packet received at node 105 and destined for node 205 is routed to node 130 according to the routing table entry.

As described above, each node has its own local topology. A method for constructing such local topologies is described below.

As noted above,  $S^k(v)$  is a  $k$ -neighborhood of a node, i.e., the set of nodes that are within  $k$  hops of that node. The following additional definition is made:

$R^k(v)$  – the  $k$ -region of a node  $v$ , which is the set of points in the two dimensional plane that are closer to node  $v$  than to any other node in  $S^k(v)$ .

Note that  $R^k(v)$  is constructed as follows. Assume that all nodes are positioned at their respective locations on the plane. Draw a straight line joining node  $v$  to some node  $u \in S^k(v)$ . Construct the perpendicular bisector of this line. This perpendicular bisector represents a half plane where node  $v$  lies in one half space. Let this half space be represented by  $P_{vu}$ . Note that if node  $w \in P_{vu}$ , then  $w$  is closer to  $v$  than to  $u$ . This process of constructing  $P_{vu}$  is repeated for every  $u \in S^k(v)$ , and  $R^k(v)$  is the intersection of the half-spaces. It can be shown that there is loopless delivery of packets using GRA if, and only if, there is no node  $v \in V$  for which there exists a  $w \in V$  such that  $w \in R^k(v)$ . An illustrative example of a  $k$ -region for node 105 of FIG. 1 is shown in FIG. 6, which in this example is a 2-region,  $R^2(105)$ . Given this condition, a flow chart of a method for use in a node for computing a local topology is shown in FIG. 7.

It is assumed that each node of the ad-hoc network performs the method of FIG. 7 every second to continually update, or create, its local topology anew. (Faster, or slower



rates may be used depending on the mobility of the nodes of the ad-hoc network.) At a high level, each node first constructs point-to-point links to a subset of nodes within hearing distance using location information – thus, determining its direct neighbors (represented by steps 605 and 610). Then, each node propagates its direct neighbor information through limited flooding to enable each node to construct its  $k$ -neighborhood,  $S^k(v)$ , for a predefined value of  $k$  as represented by step 615. (As noted above, it is presumed that each node uses the same value of  $k$ .) Thus, a local topology is formed for a reference node.

In particular, each node uses a topology sensing scheme in step 605. In this topology sensing scheme, each node periodically (or continually) broadcasts an omnidirectional pilot signal modified to additionally convey location information to any node within its transmission radius,  $r$ . (As noted above, it is assumed that two dimensional GPS coordinates are provided by each transmitting node and it is these two dimensional GPS coordinates additionally transmitted in the pilot signal.) In the context of step 605, each node listens for pilot signals transmitted by other nodes within hearing distance and recovers the GPS information for each received pilot signal for storage in a table such as the location table of FIG. 3. Thus, in step 605 each node collects GPS information for potential neighboring nodes. (Although the particular form of the omnidirectional pilot signal is not necessary for the inventive concept, for those readers interested, an illustrative omnidirectional pilot signal is described in the above-referenced, co-pending, commonly assigned, U.S. Patent application of Ahmed et al., entitled “A Topology Sensing Scheme for Networks with Mobile Nodes.”) In step 610, each node applies computational geometry to the collected GPS information to select those surrounding nodes that facilitate geometric routing and sets up point-to-point links with the selected nodes (becoming direct neighbors) and forms a direct neighbor table. (An illustrative direct neighbor table is shown in FIG. 8 for node 105 of FIG. 1.) Illustratively, there are at least three ways a node can construct its direct neighbor table using the collected GPS information.

#### Construction One:

Nodes  $u \in V$  and  $v \in V$  form a link if and only if there exists a circle with  $u$

and  $v$  on the circumference that does not contain any other node  $w \in V$ .

Construction Two:

Nodes  $u \in V$  and  $v \in V$  form an edge if and only if there exists a circle with  $u$  and  $v$  on the diameter that does not contain any other node  $w \in V$ .

5 Construction Three:

Nodes  $u \in V$  and  $v \in V$  form an edge if and only if the intersection of the circles with radius  $D_{uv}$ , one centered at  $u$  and one centered at  $v$  does not contain any other nodes  $w \in V$ .

It can be shown that if any of these three constructions is not connected, then no  
10 connected network can be formed. Construction One results in a network that is  
*1-routable*. In other words, the network constructed by construction one results in a  
network where the local neighborhood of any node is the set of nodes that are directly  
connected to it. If GRA is used to route on this network where the local neighborhood is  
the *1-neighborhood*, then any node can send packets to any other node. Network  
15 constructions two and three result in sparser networks (the number of links is lower than  
construction one). From simulation experiments, it can be shown that these networks are  
also almost *k-routable*, e.g., for  $k = 2$ , or  $k = 4$ .

After forming links with those nodes within hearing distance that meet one of the  
above-described criteria, each node through limited flooding propagates its link  
20 information (i.e., its direct neighbor table) to enable all nodes to construct their  
*k-neighborhood* in step 615. (Again, it is assumed that all nodes use the same value of  $k$ .)  
For example, and referring briefly back to FIG. 1, for a *2-neighborhood*, node 105  
receives the direct neighbor lists from nodes 110, 130 and 150 to construct the local  
topology table of FIG. 2. (It can be observed from FIGs. 1 and 2 that since node 125 was  
25 the direct neighbor to node 130, a packet received at node 105 and destined for node 125  
is routed by node 105 to node 130.) Similarly, if  $k$  was equal to three, then direct  
neighbor information is further propagated through limited flooding (e.g., node 105 would  
also receive the direct neighbor tables of nodes 115, 120, 125, 135, 140 and 150). For  
example, node 105 transmits its direct neighbor table along with a "time-to-live" field.  
30 The value of the time-to-live field is used to flood, or propagate, the direct neighbor table

information of node 105 to a limited neighborhood. Each node that receives the “time-to-live” field and the direct neighbor table of node 105, decrements the value of the “time-to-live” field. As long as the value of the “time-to-live” field is greater than zero, that receiving node further transmits the direct neighbor table of node 105 to its direct neighbors (with the decremented value of the “time-to-live” field). However, when the value of the “time-to-live” field reaches zero, that receiving node does not further propagate the direct neighbor table of 105. Although not described herein, it can be mathematically shown that the above described methods for creating a local topology generate no loops in the routing.

As noted above, it was assumed that each node knows the location (exact, or approximate) of all other nodes within a transmission radius,  $r$ . However, as noted above, it may be the case that a node is outside of the transmission range of a distant node and, therefore, cannot receive location information from that distant node. Although one alternative is simply to drop packets if the location of the distant node is not found, an alternative location update mechanism can also be used. For example, a *lazy update* mechanism may be used in which position information is periodically updated.

In this lazy update mechanism, each node maintains a list of the locations of all known nodes along with a time stamp as to when that information was generated by those nodes. Let  $p(i, k)$  be the position of node  $k$  as “seen” by node  $i$  and  $s(i, k)$  be a “time-stamp” at which the positional information was generated at node  $k$ . The time-stamp provides a vehicle for determining the age of the position information. (As can be observed from the discussion above,  $p(i, k)$  is a variation of  $l(v)$  and is two dimensional GPS information. Illustratively,  $s(i, k)$  is an integer value determined as a function of the month, day, year and time-of-day (using a 24 hour clock, e.g., 3:00 PM is 1500 hours).)

The location table of FIG. 3 is modified to include the time-stamp field as shown in FIG. 9, where the reference node,  $i$ , is node 150 of FIG. 1. For completeness, the table of FIG. 9 includes entries for node  $i$  itself (here, represented by node 150). This list of position and timestamps at a node  $i$ , is referred to as the location list, or location table,  $L(i)$ , at node  $i$ .

In accordance with the lazy update method, each node periodically transmits its

position to its direct neighbors (or, alternatively, to all nodes in its local topology) once every  $t_1$  seconds. Further, once every  $t_2$  seconds, each node transmits its location list  $L(i)$  to its direct neighbors (nodes within one hop). A flow chart of a lazy update method is shown in FIG. 10 for use in a receiving node,  $j$ . Let the receiving node  $j$  be a direct neighbor of node  $i$ . In step 905, the receiving node,  $j$ , receives location information  $p(i, k)$  from all nodes that are its direct neighbors. In step 910, receiving node  $j$ , updates its location list  $L(j)$  to reflect the current position and time-stamp for its direct neighbor nodes. (At this point it is presumed that the time-stamp information is more recent than previous local topology location transmissions stored in  $L(j)$ .) In step 915, node  $j$  receives the location lists,  $L(i)$ s, from direct neighbor nodes. In step 920, node  $j$  adds and/or modifies entries in its location list  $L(j)$  by performing the following computation for each node  $k \in V$  listed on each of the received location lists (effectively “merging” the various location lists):

If  $s(i, k) > s(j, k)$  then  $s(j, k) = s(i, k)$  and  $p(j, k) = p(i, k)$ ;

Else,

do nothing.

Also, if node  $j$  receives from a node a time-stamp associated with a node,  $k$ , not on its location list, then, by definition,  $s(i, k) > s(j, k)$ , and node  $j$  adds this new node,  $k$ , to its location list. Similarly, node  $j$  updates location information for a node,  $i$ , already listed on its location list if the received time stamp from a particular location list is more current than the existing time-stamp, i.e.,  $s(i, k) > s(j, k)$ . On the other hand, if node  $j$  has more current information for a particular node  $k$ , i.e.,  $s(i, k) \leq s(j, k)$ , then no change to the location list is performed. Thus, location information is gradually propagated throughout the ad-hoc network by the transmission of location lists from one node to its direct neighbors. This lazy update procedure results in significantly less routing overhead than flooding the entire network with the position information, whenever the position information changes significantly.

As can be observed from the above, there is a certain “warm-up” time for the ad-hoc network when using a lazy update mechanism during which some nodes will not have any information about distant nodes. As noted above, one option for the GRP routing

method is to simply discard packets if the location to a distant node is not yet known.

Also, it should be noted that it is possible for a loop to occur using a lazy update mechanism. For example, let  $t$  be the destination node for a given packet, and let node  $v$  receive this packet from a node  $u$ . If node  $v$  determines that the next hop for the packet is  $u$ , this results in a loop. In order to avoid this, when this situation happens, nodes  $u$  and  $v$  exchange  $p(u, t)$ ,  $p(v, t)$ ,  $s(u, t)$  and  $s(v, t)$ . The location of node  $t$  is resolved in favor of the node that has the more recent information. Both the nodes use this piece of information. With this modification, it can be shown that there can be no infinite loops in the route.

Turning briefly to FIG. 11, a high-level block diagram of a representative node 905 for use in the ad-hoc network of FIG. 1 is shown. Node 905 is a stored-program-control based processor architecture and includes processor 950; memory 960 (for storing program instructions and data, e.g., for communicating in accordance with the above-mentioned geometry-based routing protocol and storing location tables, etc.); communications interface(s) 965 for communicating with other nodes of the ad-hoc network via communication facilities as represented by path 966; and GPS element 970 for receiving GPS location information. Node 905 is also referred to as a router.

As described above, the inventive concept present a simple routing protocol to route packets in ad-hoc networks – large or small. The foregoing merely illustrates the principles of the invention and it will thus be appreciated that those skilled in the art will be able to devise numerous alternative arrangements which, although not explicitly described herein, embody the principles of the invention and are within its spirit and scope. For example, although the GRP identifies the closest node to a distant node, the GRP could be modified to identify any node that is closer to the distant node than the reference node. Also, although described in the context of a wireless application, the GRP could be used in other forms of packet networks such as wired networks, or networks that have combinations of wired and wireless links.